

⑫ 公開特許公報 (A)

昭61-289746

⑬ Int. Cl. 4

識別記号

庁内整理番号

⑭ 公開 昭和61年(1986)12月19日

H 04 L 11/20  
G 06 F 13/00  
15/16

1 0 2  
3 5 5

A-7117-5K  
6549-5B  
A-2116-5B

審査請求 未請求 発明の数 2 (全20頁)

⑮ 発明の名称 ユーザー・パケットの経路を選択する方法と相互接続回路

⑯ 特 願 昭61-137259

⑰ 出 願 昭61(1986)6月12日

優先権主張 ⑱ 1985年6月13日 ⑲ 米国 (U S) ⑳ 744583

⑳ 発 明 者 グ格拉斯 エイ. ジョ アメリカ合衆国テキサス州カーロルトン, ビー. オー. ボ  
ンソン ツクス 116536

㉑ 出 願 人 テキサス インストゥル アメリカ合衆国テキサス州ダラス、ノースセントラル エ  
メンツインコーポレイ クスプレスウェイ 13500  
テッド

㉒ 代 理 人 弁理士 浅 村 皓 外2名

明 細 書

1. 発明の名称

ユーザー・パケットの経路を選択する方法と相互接続回路

2. 特許請求の範囲

(1) 各々複数個の出力線及びパケット記憶装置を持つ様な複数個の節を持つ相互接続回路の中で、夫々宛先節を持つユーザー・パケットの経路を選択する方法に於て、

第1の節で多数のパケットを受取り、

前記パケットを記憶し、

記憶されている少なくとも1つのパケットの宛先節を感知し、

該記憶されているパケットに対し、宛先節までの節移動距離が最短である好ましい出力線を設定し、

前記パケットをこうして決定された好ましい出力線を介して伝送する工程を含む方法。

(2) 各々の節が入力線及び出力線によって複数個の他の節に接続される様な複数個の節を持ち、該

節が、夫々宛先節を持つ複数個のユーザー・パケットを受取る手段と、受取ったパケットを記憶する手段と、記憶した各々のパケットの宛先節を感知する手段と、記憶されている各々のパケットに対し、該記憶されているパケットの宛先節までの節移動距離が最短である通路に接続された好ましい出力線を決定する手段と、前記パケットを前記好ましい出力線を介して伝送する手段を持つ様なユーザー・パケットを伝送する為の相互接続回路。

3. 発明の詳細な説明

産業上の利用分野

本発明は全般的に多節節計算機相互接続回路網(ネットワーク)、更に具体的に云えば、経路伝送効率を高くすると共に、回路網のグリッドロックの問題を最小限に抑える為、この様な回路網でパケットの経路を選択する方法と装置に関する。

従来の技術及び問題点

計算機システムはメモリ、プロセッサ(CPU)及び入力/出力装置の様な多数の別々の部品を含

んでいる。これらの全ての部品は互いに連絡することが出来なければならない。相互連絡の為に最も普通に使われている従来の手段は、バス及びクロスバーを使うことであった。

電子産業の最近の進歩により、バス及びクロスバー相互接続方式でも不十分になりつつある。大規模集積(VLSI)により、1個のチップの上に小形の低速プロセッサを形成することが可能になっている。従って、多くの単一チップ・プロセッサを構成することにより、大形の高速プロセッサを構成するのがコスト効果がある。然し、こういうことをすると、プロセッサの間の相互接続が、プロセッサの全体的な効率を高めようとする時の主な障害になる。

理想を云えば、相互接続は、十分な帯域幅を持つと共に、非常に多数のプロセッサを妥当なコストで接続することが出来なければならない。更に、システムの能力を高める為にプロセッサを追加することにより、相互接続を拡張し得ることが非常に望ましい。別のプロセッサ又は装置を用いて回

路網を拡張する際、過度のコストを伴わずに、且つ将来の拡張を見込んで最初是不変であるハードウェアを購入することを必要とせずに、相互接続の帯域幅がそれに伴って拡大することが望ましい。

従来のバス及びクロスバー相互接続方式は、こういう条件に合わない。バスは、それがサービスし得るプロセッサの数に慣習的な制約しかないが、帯域幅が一定である。クロスバーの帯域幅はそれが相互接続するプロセッサの数と共に増加するが、そのコストはプロセッサの数の自乗に比例して増大し、最初に大きめに設計しておかなければ、まったく拡張することが出来ない。

地域の計算機回路網(ローカルエリアコンピュータネットワーク)では、帯域幅の条件がかなり小さく、1つのバスで足りるのが普通であるが、バスは故障に対する許容度がない。バスの故障により、回路網全体が故障する。クロスバーは、コストが $N^2$ で増大する為に、地域の計算機回路網には一般的に実用性がない。クロスバーの故障に

対する許容度は、クロスバーの内部の各節の節の間に1つのリンクしかないから、バスの故障に対する許容度より遙かに大きいだけである。従って、リンクの故障により、クロスバー内の2つの節の間の通信が出来なくなる。クロスバー接続方式もバス接続方式も、上に述べた理由で、地域の計算機回路網を容易に拡張することが出来ない。

更に最近になって、回路切換え方式ではなく、パケット切換えを用いることにより、拡張能力及び故障に対する許容度が不完全ながら幾分達成される様になった。パケット切換え方式では、ユーザーのメッセージを分離(分解)し、多数のユーザー・パケットに分けて伝送する。パケットはユーザー・データ及び制御データを含んでいることがあり、その全てが複合体として切換えられる。一般的に、ユーザー・データの各々のパケットは見出しを持ち、これがパケットの最終的な宛先に関する情報を持っている。

パケット切換えは、パケットの起点とその最終宛先の間に幾つかの交替的な経路を持つ多重節シ

ステムに使われる場合が多い。パケットの切換え及び多重通信経路を使うことにより、故障したスイッチ又は線路を側路することが出来るし、パケット切換え制御装置が各々の節に分布している場合、最初に大きめに設計しておかなくても、システムを拡張することが出来る。然し、こういう節の制御ソフトウェアは一般的に回路網の故障に適応する際、融通性がない。

多重節相互接続システムにおけるパケットの切換えは、この他にも問題がある。その1つは、負荷の平衡である。パケットが、その宛先に達するまでの節移動距離が最短である様な経路で送られる様な経路選択規則を選んだ場合、ある経路は他の経路よりもずっと高い頻度で選択され、システムの負荷に不平衡が生じ、静的なデッドロックの慣れがある。多重節パケット切換え方式に起こる別の問題は動的なデッドロックである。動的なデッドロックは、ある節が、別のパケットを受取る為にパケット記憶バッファを空ける為に、あるパケットを「強制的に」その節から出す時に起こる。

このバケットはある出力線に沿って押出されることがあるが、この出力線がこのバケットをそのバケットの宛先から遠ざかる向きに移動することがある。経路選択規則の一組がよく出来ていないと、バケットが無限に循環することがあり、その結果動的なデッドロックになる。

従って、節の間に最低の論理的な距離を持ち、静的及び動的なデッドロックを防止する様な一組の経路選択規則によって操作される多重節バケット切換え回路網に対する要望が生じている。更に、故障に対する許容度が極めて高く、バケット切換え機能について自己修理形の分布した制御節を持つバケット切換え回路網に対する要望がある。

#### 問題点を解決する為の手段及び作用

本発明は多重節計算機相互接続回路網でユーザ・バケットの経路を選択する方法と装置を説明する。各々のユーザ・バケットはその宛先節を表わす識別子を持っている。各々の節が入力線及び出力線により、複数個の他の節に接続されている。各々の節が複数個のリンクを持ち、各々のリ

ンクが入力線に接続されると共に、多数のバケット記憶バッファを持っていて任意の出力線に選択的に接続することが出来る。

動作の際、節がその入力線に多数の情報バケットを受取る。情報バケットが夫々バッファに記憶される。次に、節が各々のバケットの宛先を感知し、ルックアップ・テーブル又は経路選択テーブルを見ることにより、各々のバケットに対する好ましい出力線を選択する。好ましい出力線は、宛先節までの節移動距離が最短である経路の一部と定義する。一旦、各々のバケットに対する好ましい出力線及びそれに対応する節移動距離が決定されたら、各々のリンクのバッファの中で、節移動距離が最短である1つのバケットを選択する。その後、各々のリンクから1つずつのこのバケットが、それに関連する節移動距離の昇順で、そういうバケットに対する出力線が既に割当てられていなければ、夫々の好ましい出力線に割当てられる。

例れかの出力線がこの後バケットに割当てられ

ないまゝである場合、節は、それまでに割当てられたバケットの伝送の後、どのリンクが依然として一杯であるか、即ちまだ空のバッファを持っていないかを視察する。一杯と見込まれる各々のリンクに対しては、割当てられていない線を使いきるまで、又は全てのリンクが最早一杯と見込まれなくなるまで、記憶されているバケットを出力線に割当てる。上に述べた工程の後に、まだ割当てられていない出力線があれば、割当てられていないで残っているバケットは、割当てられていないバケットに対する好ましい線である様な、割当てられていない出力線が残なくなるまで、夫々の好ましい出力線に割当てることができる。この様にして記憶されているバケットが選択された後、割当てられたバケットを伝送する。

#### 実施例

本発明の経路選択装置は任意の多重節相互接続回路網に使うことが出来るが、ハイパートロイダル形相互接続方式を用いることが好ましい。この明開書で云う「ハイパートロイダル形」とは、各

々の節が2n本の接続部を持ち、n個のリングの1メンバである様な種類の回路網の構成を云う。リング次元又は回路網のリングがその中に入っている考えとしてのリング群の数もnである。

第1図は1リング次元、即ち、 $n=1$ のハイパートロイドを示す。各々の節10は2( $=2n$ )組の入力/出力線12しか持たず、1( $=1n$ )リングだけの1メンバである。1次元のハイパートロイドは、バケットの切換えの為に交差的な経路が1つしかないで、少数の節にだけ用いられる。

第2図は9個の節及び2リング次元のハイパートロイダル形回路網を示す。本発明は3又は更に多くのリング次元及び多数の節を持つハイパートロイダル形回路網に使うことも出来るが、本発明を説明する為に第2図に示す構造を使う。各々の節10には説明の便宜の為に大文字のアルファベットが記入されているが、実際には節の確認は勿論デジタル順序によって定められる。図示の2次元の場合、各々の節10が4本の全2重線12

によって隣接する節に接続されている。各々の線12が2つの節を互いに接続すると共に、説明の便宜上、二重に符号がつけられている。即ち、節Aでは、節Aを節Bに接続する線が線a3であると共に線b1でもある。各々の節10は2つのリングの1メンバであり、各々のリングが異なるリング次元にある。節Aはリング14及びリング16の1メンバである。リング14が節B及びCを含み、リング16が節D及びGを含む。

ハイパートロイドは他の回路網に較べて多数の利点がある。これは節の間の距離を短く保ちながら、高度の拡張能力がある。ハイパートロイド形構造は節の間に多重経路を作り、こうして自動的な負荷の平衡をとること並びにフェールソフト能力が得られる。シミュレーションによると、ハイパートロイドは、所定数の節及びリンクに対して最大の帯域幅が得られる回路網であることが判った。

発信節と宛先節の間のハイパートロイド形回路網の平均論理(即ち)距離が比較的短いことは、

る。バッファ38乃至44の動作が制御バス53を介して、バッファ制御装置51によって制御される。CPU48は、マイクロプロセッサであってよいが、アドレス・バス50を介してMUX/DX制御装置47に接続される。アドレス・バス50はCPU48からバッファ制御装置51に対して命令をも送出す。

第4図はリンク34を詳しく示している。制御バス49がMUX/DX制御装置47(第3図)からの命令を4者択1直列デマルチプレクサ36に伝える。この命令に従って、デマルチプレクサ36が入力線18の到来パケットを4つのバッファ入力線62、64、66、68の内の1つに切替える。これと平行して、バッファ制御装置51(第3図)が、夫々の入力線62乃至68に現れるパケットを受取る様に、バッファ38乃至44の内の1つに知らせる。

バッファ38の内部構造が第4図に詳しく示されている。制御装置70が制御バス53からの命令を受取り、それ自身が制御線78乃至82を介

して配送時間及び最大配送帯域幅の両方に強い影響がある。ハイパートロイド形回路網を拡張するにつれて、節の間の平均距離はゆっくりと増大する。2次元回路網で節の数を2倍にしても、平均節間距離は $\sqrt{2}$ 倍にしか増加しない。

第3図は第2図に示した回路網内の節Aの内部構造の回路図である。第2列では、節Aが、線a1、a2、a3、a4を含む一組の全2進線12により、周囲の節に接続されている。第3図は、線a1、a2、a3、a4が更に夫々入力線18、20、22、24と出力線26、28、30、32とに分れることを示している。入力線18乃至24及び出力線26乃至32が、図面では判り易くする為に分離して示されている。

入力線18がリンク34に接続される。このリンクはデマルチプレクサ36、複数個のバケット記憶バッファ38乃至44及びマルチプレクサ46で構成される。マルチプレクサ/デマルチプレクサ制御装置47が制御バス49を介してマルチプレクサ46及びデマルチプレクサ36を制御す

して素子72乃至76を制御する。線78の信号が直列入力/並列出力シフトレジスタ72を付能して線62のバケットを受取る様にし、これが直列形式で節の間に伝送される。素子72はバッファ・バス84にバケットが並列形式で利用出来る様にする。制御装置70が、線80の信号により、バケットの経路選択の見出しをバッファ74にラッチすることが出来る様にする。バッファ74は、宛先節(ユーザー・バケットにある)又は発信節及び節距離数(節距離バケットにある)の様な項目を含む見出しが、CPU48によって読取られる様に、データ・バス118に得られる様にする。

並列入力/直列出力シフトレジスタ76が、制御装置70から制御線82を介して送られる信号によって付能された時、バケットを直列形式に変換し、それをバッファ直列出力線86から伝送する。制御装置70がレジスタ72を付能して、レジスタ76が線86から出のバケットを伝送することが付能されると同時に、到来バケットを受取る事が出来る様にする。線86が4者択1マ

ルチプレクサ46に接続され、これが制御バス49から受取った命令に従って、出力線86乃至92の内の1つを内部出力線54に切換える。

バッファ40乃至44の構成はバッファ38の構成と同じである。

第3図に戻って説明すると、マルチプレクサ46が内部出力線54の出のバケットをクロスバー122に伝送する。入力線20乃至24がリンク56、58、60のマルチプレクサ(図面に示していない)に接続される。リンク56乃至60はリンク34同じ様に構成されている。リンク58乃至60からの出のバケットが内部出力線110、112、114を夫々介してクロスバー122に伝送される。

CPU48がデータ・バス118を介してランダムアクセス・メモリ(RAM)116と連絡する。データ・バス118は、CPU48と、MUX/DX制御装置47、主バッファ制御装置51、CPU出力インターフェース素子94及び局部CPUインターフェース96との間の連絡が出来る

は114の内のどれを外部出力線26、28、30又は32と電気接続すべきかについて、クロスバー制御装置98に指示する。こういう命令が制御装置98からクロスバー・バス120を介してクロスバー122に中継される。

CPU出力インターフェース94は、CPU48が直列メッセージ線100を介して他の節と「ハンドシェイク」することが出来る様にする。インターフェース94が、リンク34、56乃至60の内の1つにバケットを受取った時、適当な出力線に確信信号又は誤りメッセージ(受取ったバケットが誤り検査によって、不良であることが判る場合)を伝達する。送信側の節がインターフェース94から確信を受取らないことにより、後で説明する故障検出手順が設定される。

回路網にある各々の節A乃至Iには1つ又は更に多くの局部(ローカル)装置124が接続されている。これらの装置はプロセッサ、メモリ又は入力/出力装置であってよい。装置124が節10にインターフェース126によって接続される。

様にする。

CPU48がバッファ74から見出し情報及び同様なエレメントを順次読取って、どのユーザー・バケットがその宛先に最も近いかを決定すると共に、どの出力線26乃至32が宛先節に対して最短の経路であるかを決定する。CPU48は、RAM116に記憶されているルックアップ・テーブル(後で説明する)の助けを借りて、こういう比較を行なう。バッファ74等に対して書込む能力が、回路網の動作の始めに、ルックアップ・テーブルの作成に関連して、増分距離バケットを伝送するのに必要であるが、これは後で説明する。

CPU48がリンク34に記憶されている1つのバケット、リンク56に記憶されている1つのバケット、リンク60に記憶されている1つのバケット及びリンク48に記憶されている1つのバケットを、システムの次のクロックパルスで伝送する為を選択する。CPU48はアドレス・バス50により、内部出力線54、110、112又

インターフェース126が局部装置出力線128及び局部装置出力線130を介してクロスバー122に接続される。CPU48は、局部CPUインターフェース96からバス118を介して局部的に発生されたバケットの宛先節の確認を受取った時、インターフェース126の内容を任意の出力線26乃至32を介して伝送する様にクロスバー122に命令することが出来る。CPU48は、A(自分の節)に等しい宛先節を持つ記憶されているバケットを線128を介してインターフェース126に伝送する様に、(制御装置98を介して)クロスバー122に命令することも出来る。インターフェース96が、バス102を介して伝送すべきバケットの宛先節を受取ると共に、送信準備完了信号及びその他の情報を装置124から線104、106を介して受取る。局部装置124及びインターフェース126がバス109によって接続されている。

この発明は2次元のハイバートロイダル形の組合について説明しているが、節10の構造は、追

加のリング次元を受入れる様に容易に拡張することが出来ることをもう一度述べておきたい。追加の各々のリング次元は、追加の2本の入力線、追加の2本の出力線及び追加の2つのリンクを必要とする。節10の残りの部品は機能的に同じである。各々のリンクにあるバッファの数を増減することが出来るが、1つのリンク当たり少なくとも4つのバッファがあることが好ましい。ハイパートロイドの次元の数及びデータ・バケット記憶バッファの数は、所望の帯域幅、回路網の利用度及び全体的な節間の伝送速度に従って選択される。

第5図はRAM116に記憶されているテーブル作成ソフトウェアのフローチャートである。システムを動作させるのに十分なルックアップ・テーブルはPROM又は同様な装置に結線にしてもよいが、システムの1つのリンク又は節が故障した場合、或いはこの回路網に対して追加をした場合、修正することが出来る様に、ルックアップ・テーブルを持久形メモリに記憶することが好ましい。工程132で電源を投入した後、シ

ステム内の各々の節が工程134で節距離バケットを発生する。各々の節距離バケットが節同定符号及び節距離数を含む。第5図は第2回の節Aに対するルックアップ・テーブルの作成を示す。Aで発生された節距離バケットの同定部分は“A”である。節距離バケットを発生する時、節距離数は0である。

節距離バケットを発生した後、発生した節が工程136で節距離バケットを各々のリンク34、56、58、60(第3図)のバッファに入れる。工程138で、節が節距離バケットを各々の出力線を介して隣接する接続された節に伝送する。節距離バケットの伝送と同時に、節Aが工程140で、入力線を介して隣りから4つの節距離バケットを受取る。この工程を判定として示してあるが、これは、テーブル作成過程の後の時点で、節Aが何等節距離バケットを受取らないことがあるからである。第5図では、受取った節距離バケットを全体的に $X(n)$ で表わしてある。ここで $X$ は発信節の同定符号であり、 $n$ は受取った節距離バケ

ットの節距離数である。

何等かの節距離バケットを受取った場合、CPU48(第3図)が、工程142で、1番目のリンクから開始して、それらを直列に読取る。工程144で、CPU48は、受取った節距離バケットが自分で発生したものかどうかを尋ねる。 $X$ がAに等しければ、CPU48は工程146でこの節距離バケットを廃棄する。 $X$ がAに等しくなければ、工程148で節は節距離バケットの節距離数を定数、好ましくは1だけ増加する。バケットはそれが現れた入力線とも関係している。

判定工程150で、CPU48が、RAM116(第3図)にあるルックアップ・テーブルの、節の同定符号及び入力線 $l$ に対応する項目を見る。全体の内の $T(X, l)$ にある項目は、出力線 $l$ を含む経路で、節 $X$ までの節 $l$ の距離に対応する。この項目が0であって、この項目に対応する情報をまだ受取っていないことを表わす場合、工程152でこの項目が $(n+1)$ に設定される。項目 $T(X, l)$ がゼロ以外であれば、工程154で、

バケット $(n+1, l)$ の節距離数 $(n+1)$ をテーブルの対応する項目 $T(X, l)$ と比較する。節距離数がテーブルに存在する項目よりも小さければ、工程152で、 $(n+1)$ をこの項目に等しく設定する。そうでなければ、工程146で、節距離バケット $X(n+1, l)$ を廃棄する。

あるバケットの節距離数 $(n+1)$ がルックアップ・テーブルの項目に設定される場合、節距離バケット $X(n+1)$ を線 $l$ から切断し、保存して、工程156で伝送キューに入れる。増分した時、節距離バケットが全ての線を介して隣接する節に伝送され、こうして $X(n+1)$ が辿る経路では節 $X$ がどのくらい遠いかに関する正しい情報を伝える。他方、節距離バケット $X(n+1)$ が、テーブルの節 $T(X, l)$ の所に既に存在する値よりも小さい節距離数 $(n+1)$ を持つ場合、これは、 $X(n+1, l)$ が辿る経路が線 $l$ を含む、節 $X$ までの最短経路ではないことを示している。従って、節距離バケットを廃棄する。

工程156で、 $X(n+1)$ が、バッファに直

接的に入れる代りに、キューに入る。これは、テーブル作成過程の間、出力線を通じて伝送すべき節距離バケットの数が、たちまち、データを記憶する為に利用し得るバッファの数を越えるからである。例えば、システムの最初のクロック・サイクルの間、節Aが節距離バケットD(0)、G(0)、D(0)及びC(0)(第2図参照)を受取る。これらの節距離バケットの節距離数を1だけ増数した後、節距離バケットD(1)、G(1)、D(1)及びC(1)になる。1の入力がルックアップ・テーブルのT(D, 2)、T(G, 4)、T(D, 3)及びT(C, 1)に入られる。最初のクロック・サイクルの間に受取った全ての節距離バケットが、ルックアップ・テーブルに情報を供給される為に使われるから、これらの節距離バケットはどれも廃棄されず、従って全て再び伝送される。4回繰返されると、最初のクロック・サイクルの間に受取った節距離バケットだけで、再伝送を持つ為に、利用し得る16個の記憶バッファ全部を占める、従って、1より

る様に作る。この為、 $X(n+1)$ が伝送される前に4回繰返される。その後手順は工程168で、伝送キュー内にこれ以外の節距離バケットがあるかどうかを尋ねる。あれば、工程170が、このキューに入っている別の節距離バケットを保持する為に残っているバッファ(BUFF(ℓ, a))があるかどうかを決定する。バッファは、1つのリンクに対して1つずつ、一度に4個埋ったり空になるから、空のバッファの数は4の倍数であり、“a”はバッファ番号であり、1から4まで変化する。伝送キュー内にそれ以外の節距離バケットがない場合、又はこの様な節距離バケットを受容する為の別の空のバッファがない場合、各々のリンク(ℓ)内のバッファ(a)の中味が工程172で全てで隣接する節に伝送される。別の空のバッファがあれば、工程174で、その中に別の節距離バケットが入られ、手順はループ状に工程168に戻る。

工程164で、伝送キュー内に節距離バケットが見付からない場合、手順が工程176にプラン

大きいこの後の有効な(廃棄されない)節距離バケットは、バッファが空のを待つてキューに記憶される。

工程158で、受信リンクℓを1だけ増数し、工程160で、線の番号を最大のリンク番号である4に対して試験する。(リンクℓ及び入力線ℓは、第3図に示す様に、関連した構成である。)1つ又は更に多くのリンクをまだ調べなければならない場合、テーブル作成手順は工程160から工程144に戻り、残りのリンクに対してこの過程を繰返す。全てのリンクを探索したら、次に手順は工程164(図面の右側)に進み、伝送キュー内に増数した節距離バケット( $X(n+1)$ )が存在するかどうかを尋ねる。判定工程140からの「ノー」ブランチもこの順にブランチする。大抵の手順の場合がそうであるが、伝送キュー内に $X(n+1)$ が存在する場合、工程168で、キューの一番上にある節距離バケット $X(n+1)$ を各々のリンクの1つのバッファに入れ、各々の出力線で節距離バケットを再伝送することが出来

る。この工程は、回路網の中で何れかの有効な節距離バケットが依然として伝送されているかどうかを見付ける部分的な手順を開始する。工程176が、何れかのリンクの何れかのバッファに節距離バケット $X(n+1)$ が存在するかどうかを尋ねる。1つ又は更に多くの節距離バケットが再伝送が出来る状態にあれば、工程178でカウント変数を0に設定し、工程172で、見付かって節距離バケットを再伝送する。バッファに節距離バケットが見付からなければ、工程180でカウントを1だけ増数して、任意に設定した定数Sと比較し、節が節距離バケットを再伝送せずにどれだけのサイクルが経過したかを判定する。Sは、キューの支線を考慮に入れた時、最大の論理的な隔たりを持つ節から全ての線を介して伝送された節距離バケットが主体の節(今の場合節A)に到達することが出来る様に設定される。工程182で、カウントがSより小さければ、手順は工程140に戻り、テーブル作成手順の次のサイクルを開始する。工程172で4個1群のバッファの中

味を伝送した後も、手順は工程140に戻る。工程182で、カウントがSより大きいとか又はそれに等しくて、即距離バケットがある数のサイクルの間伝送されていないことを示す時、工程186で「A一杯」バケットを伝送し、節Aのルックアップ・テーブルが最終的に決ったことを示す。工程188が、この節がシステム内の他の各々の節から同様な「B一杯」、「C一杯」等を受取ったかどうかを試験する。そうならない場合、手順は工程140に戻って別のサイクルを開始する。然し、システム内の各々の節から「X一杯」バケットを受取った場合、これは、回路網内の全てのルックアップ・テーブルが最終的に決ったことを示しており、回路網は工程190でユーザー・バケットの切換えを開始する用意が出来ている。この時、第7図に示すバケット切換え手順を用いる。即距離バケットの伝送工程172で、バッファ“a”は順次選択することが好ましい。即ち、伝送の最初のサイクルで、各々のリンク1乃至4に対するバッファ番号1が選択される場合、二番目

のサイクルでバッファ2が伝送の為に選択されると言う様に作る。どの宛先節バケットも、それを受取った順序から外れて伝送されない様に保証する為に、このパターンが繰返される。最短の論理的な経路をたどる節距離バケットが節Aに最初に到着し、更に順次通した経路を辿る節距離バケットは後になって到着するから、この為にテーブル作成の効率が高くなる。節距離バケットをそこから受取った節に対して、節距離バケットを送り返さないことにより、伝送の記憶を幾分改善することが出来る。これは、こうして再伝送する節距離バケットが単に起点の節によって廃棄されるだけであるからである。然し、この様な論理的な調整は、実行するのに別のオーバーヘッドを必要とし、その為、多くの場合には入れることは望ましくないことがある。

第6図はRAM素子116(第3図)に記憶される完成されたルックアップ・テーブル192の概念図である。第6図に示すルックアップ・テーブルは、第2図に示した比較的簡単な2次元のハ

イパートロイダル形回路網に対応しており、追加の次元又は節を持つ回路網に対しては、勿論拡張される。完成されたテーブルは、参照した出力線に沿った、参照した節に対する最短の論理的な距離を示している。出力線28(全2重線a2の一部)から伝送され、宛先節Gを持つユーザー・バケットに対応して、項目T(G,2)が2つの節距離に等しい。項目T(G,4)は節Gがこの経路では1節距離しか離れていないから、バケットGを線1で送る方が一層経済的であることを示している。

ハイパートロイダル形回路網が論理距離の増大を最小にするのに好ましいが、任意の多重節システム・アーキテクチャに対して第6図に例示する様なルックアップ・テーブルを作成することが出来る。

各々の節に対してルックアップ・テーブル192が一旦作成されると、回路網は局部装置124(第3図)からのユーザー・バケットを回路網に切換える用意が出来る。この明細書で言う「ユー

ザー・バケット」とは、データ及び/又は命令を含む一定の長さを持つユーザー・プログラム資料のバケットを指す。ユーザー・プログラム動作の各々のバケット切換えクロック・サイクルの間、任意の節10(第2図)が入力線1~4(第3図の18, 20, 22, 24)を介してユーザー・バケットP(X)を受取り、これがリンク35, 58, 57, 60で、CPU48によって選択された空のバッファに記憶される。バケット切換え手順は、常にユーザー・バケットを受取る用意が出来た空の状態にある少なくとも4つのバッファが、1つのリンクにバッファ1つずつ、ある様に動作する。

第7図について説明すると、工程194で、ユーザー・バケットがリンク1に対して1つずつ、4つの空のバッファに入力される。次に工程196が、他の節からのユーザー・バケットを受取った後、まだ空でないバッファがあるかどうかを尋ねる。1つ又は更に多くの空でないバッファが存在すれば、局部的な起点からのユーザー・バケ



ットP(X)が、局部装置124(第3図)から1つ又は更に多くの空のパッファに入れられる。第3図に戻って説明すると、局部的なユーザー・バケットが、クロスバー122及び制御装置98の作用により、リンク34, 56, 58又は60にある1つのパッファに切換えられる。

局部的な起点を持つバケットを空のパッファに入れることが出来るかどうかにより、工程200で変数 $l$ が1にセットされる。次に工程202が、検査したリンク内の全てのパッファが空であるかどうかを検査する。リンクのパッファが全部空であれば、プログラムは工程204にブランチし、そこで工程204でリンク番号を増数し、工程206でリンク番号が最大値を超えているかどうかを試験し、工程202で順番の次のリンクを検査する。空でないリンクが突止められると、パッファ度数 $a$ を工程205で1にセットする。工程206で、パッファ・ラッチ74(第4図)からバケットの見出しの宛先節を讀取る。その後、バケットの宛先節Xに対し、ルックアップ・テ

ットが、組Sの1メンバとして選択される。組Sに属するユーザー・バケットは、隣接する節に伝送する為の出力線に割当てで、第1の優先順位を持つ。S( $l$ )の各メンバは、選ばれた記憶されているユーザー・バケットの好ましい出力線に相当する出力線 $ol(S(l))$ に関連してあり、選ばれたユーザー・バケットの節距離に相当する節距離 $d(S(l))$ を持っている。

一旦Sの1メンバが現在のリンク $l$ で選ばれると、工程204でリンク番号を1だけ増数し、工程206で節にあるリンクの総数と比較する。リンク番号が節にあるリンク総数より小さい場合、この過程を繰返して、組Sの別のメンバを選択する。

組Sのメンバの選択を完了したら、工程216(一番上)で各メンバが夫々の宛先節までの節距離 $d$ に従って、一番小さいものから一番大きいものまで分類される。工程218で、S( $l$ )の内、 $d$ が一番小さい1メンバを検索する。工程220が、検索されたユーザー・バケットに

ブルの項目T(X,  $ol$ )を $ol-1$ 乃至4に対して検索する。第7図では、 $ol$ が1乃至4の範囲の出力線度数である。次に、T(X,  $ol$ )の最小値を拾い出し、それを、パッファ( $l, a$ )にあるユーザー・バケットが出てから移動する距離(P(X,  $l, a$ ))に等しいと置くことにより、最小の論理距離を決定する。工程208で、ユーザー・バケットP(X,  $l, a$ )にとって好ましい出力線 $ol$ をケーブルTで最小距離を持つ出力線に等しいと置く。

工程210で、パッファ $a$ を1だけ増数し、工程212で、各々のリンクにあるパッファの数(今の場合は4)と比較して、考えられるパッファの数を越えたかどうかを検査する。 $a$ が4より小さければ、リンク内の各々のパッファに対して、P(X,  $l, a$ )及び $ol(P(X, l, a))$ を決定する過程を繰返す。工程212で、現在のリンクにある全てのパッファを検査したと判定されると、工程214で、 $d$ が最も小さいリンクに記憶されている1つのユーザー・バケ

として好ましい出力線がその時まで割当てられているかどうかを決定する。割当てられていなければ、工程222で、この出力線に対する伝送キューの一番上にS( $l$ )を入れることにより、この出力線が割当てられる。S( $l$ )の検索された1メンバが、既に割当てられている好ましい線を持つ場合、手順は工程224で、 $d$ が次に一番小さいS( $l$ )の1メンバを考慮する。

従って、工程216乃至224により、伝送の為それらの出力線が利用出来るかどうかに応じ、S( $l$ )の1つから全部までのメンバが出力線に割当てられる。その宛先までの節距離が一番短いS( $l$ )のメンバは確実に出力線が割当てられ、システムの現在のクロック・サイクルの間に伝送される。S( $l$ )のそれに残るメンバが出力線に割当てられるかどうかは、それらの出力線がそれまで考慮されたS( $l$ )のメンバに割当てられているかどうかに関係する。

工程226が考慮すべきS( $l$ )の残りのメンバがあるかどうかを尋ねる。あれば、手順は工程

224に戻る。なければ、手順は、判定工程228で伝送の為にユーザー・パケットを選択する次の段階に進む。この工程は、S(1)のメンバが割当てられた後、まだ割当てられていない出力線が残っているかどうかを尋ねる。割当てられないで残っている出力線がなければ、割当てられた4つのユーザー・パケットは伝送の用意が出来ており、手順は伝送工程230にブランチする。

割当てられないで残っている出力線があれば、ユーザー・パケット選択手順の第2段階は、最初に工程232で $\epsilon$ を1にセットする。この手順が工程234で現在のリンクを検査して、そのバッファの全部が割当てられていないユーザー・パケットを持っているかどうかを決定する。現在のリンクが「一杯」であれば、即ち、その全てのバッファが割当てられていないユーザー・パケットを持っていれば、工程236は、割当てられていない出力線に相当する好ましい出力線を何れかの割当てられていないユーザー・パケットが持っているかどうかを決定する。そうであれば、工程23

8で、約合ユーザー・パケットに好ましい出力線が割当てられる。工程240で、リンク番号をリンクの最大数に対して比較する。全てのリンクをまだ考慮していなければ、工程242で、リンク番号を1だけ増進し、工程234で次のリンクを考慮する。

割当てプログラムのこの段階までに全てのリンクが考慮されていれば、ユーザー・パケット割当ての第3段階が工程244から始まり、この工程は割当てられていない出力線が残っているかどうかを尋ねる。残っていなければ、手順は伝送工程230にブランチする。割当てられないで残っている出力線があれば、工程246が各々のリンクを検査して、その何れかが一杯であるかどうかを決定する。1つ又は更に多くの一杯のリンクが残っていれば、プログラムは工程248でそのユーザー・パケットの好ましい出力線が割当てられていない出力線と同じであるかどうかを考慮せずに、一杯のリンクにある割当てられていないユーザー・パケットを割当てられていない出力線に

割当てる。この段階は、次のパケット切換えクロック・サイクルで到来データを受取る少なくとも1つの空のバッファが各々のリンクに利用出来る様に保証する。工程246で、各々のリンクが1つの空のバッファ、又は既に割当てられている情報パケットを持っている1つのバッファを有するかが判定されると、手順は工程250にブランチし、そこで、それらの好ましい線が割当てられていない出力線がある場合、それに相当すれば、残りの全ての情報パケットが割当てられる。他方のブランチの工程248の後、プログラムは割当てられていない出力線が残っているかどうかをもう一度尋ねる。あれば、プログラムは工程250に進む。そうでなければ、プログラムは伝送工程230にブランチする。

工程250を完了した後、又は工程250より前に割当てられていない出力線を使い果たした場合、工程230で、割当てられた情報パケットが夫々の割当てられた出力線を介して伝送される。工程230は、入力線a1乃至a4から別の情報パケ

ットを受取るのと同様である。この為、説明の便宜上、工程230及び工程194を考への上では図1254によって隔てているが、前のパケット切換えサイクルの伝送工程230は、後のパケット切換えサイクルの工程194で情報パケットを受取るのと同時に行なわれる。第4図に戻って説明すると、こういうことが可能であるのは、バッファ38がリンク34にある唯一の空のバッファである場合、出のパケットがシフトレジスタ76から線86に直列に出力されるのと同時に、到来パケットがレジスタ72に直列に入力されるからである。この為、パケットの間で場所の争いが起こらない。

上に説明したパケット切換え手順を要約すると、次の通りである。

1. 最初に、各々のリンクで(その先頭節までの距離の点で)一番近い情報パケットが組Sを構成する。Sにあるパケットは、それに関連した距離"d"に従って、一番近いものから一番遠いものまで、1つの順序に配置する。次に、Sにあ



3)) - 2、0より(P(D、4、3)) - 3、0より(P(I、4、4)) = 1である。線3は割当てられておらず、従ってP(H、4、1)に割当てられる。リンク4の残りのバケットは既に割当てられている線を好ましいとする。

段階2の終りに、依然として「一杯」のリンク(リンク1)があり、割当てられていない線(出力線4)がある。従って、工程244及び246(第7図)がバケット選択手順を強制的に第3の状態(工程248)にする。リンク1の第1のバッファにあるバケット(P(I、1、1))を取出し、宛先1を持つバケットに対する好ましい線を考慮せずに、伝送の為に出力線4に割当てて。都合のいいことに、線4は好ましい線1と同程度により選択である(第6図参照)。

全ての出力線がバケットに割当てられたので、モードは工程230で次のクロック・サイクルの始めに伝送する用意が整う。第8例は伝送すべきバケットを丸で開んで、夫々に割当てられた出力線を一番右側の列に示してある。

である。線又は節の故障により、テーブルの中の幾つかの値が変化する(第10図及び第11図参照)。

第262で、第7図に概略的に示したバケット伝送手順の大部分を短絡する。全てのルックアップ・テーブルを発生するまで、ユーザー・バケットP(Z)は、第7図の工程248に従って、バケットの好ましい線に関係なく、一杯のリンクにある1つのバッファを空ける為に伝送されるだけである。第7図の他のバケット選択段階(214乃至222)、(234乃至242)、(250)はルックアップ・テーブルを使うから、それらを略する。

工程265でFPFLGを1にセットする。不良バケットは、工程264で、利用出来る様になった時に全ての出力線に伝送する。隣接する節が工程266で夫々不良バケットを受取り、それらは、工程263から始まる、送信側の節が行なった手順を真似する。これによって、最終的にシステム内の全てのルックアップ・テーブルが置き

第9図はシステム内の節又は線が故障した時に入る不良バケット発生サブルーチンのフローチャートである。工程の開始250で再送信増数変数Rがゼロにセットされ、不良バケット・フラグFPFLGもゼロにセットされる。工程256で、節Xが、XがYに対してバケットを伝送した後に、隣接する節Yから確認符号を受取らなかったことに気付く。これによって、節Xが工程257でバケットを再送信すると共に、工程259でRを増数する。工程261で、Rを予定の定数Kと比較する。RがKより小さければ、再送信手順は工程256にループ状に戻る。RがKより大きいか又はそれに等しければ、工程258で不良バケットFPFが発生される。

節は工程263で、FPFLG=0であるかどうかを検査する。最初のパスで、FPFLGは0に等しく、従って節は工程260でそのルックアップ・テーブルを払拭する。ルックアップ・テーブルを払拭することは、その項目が、故障前のシステムの状態に対してだけ有効である為に、必要

れる。

節が既にそのルックアップ・テーブルを払拭しているから、工程266では別の不良バケットを受取る根拠が大きい。然し、この節に対し、工程265でFPFLGが1にセットされているから、工程263で手順は工程260のルックアップ・テーブルを再び払拭する代りに、工程268にブランチする。工程268では、受取った不良バケットを廃棄し、再び伝送しない。このブランチは最終的にはシステム内の全ての不良バケットを除く。

工程264で節が不良バケットを伝送した後、第5図に示したテーブル作成手順に従って、工程132でそのルックアップ・テーブルを再び作成し始める。工程270で、FPFLGがゼロに戻され、システムの1クロック・サイクルの後にサブルーチンに備える。システムがこの節の近くにある全ての現在の不良バケットを廃棄する様にする為、この工程は1サイクル後に行なう。全てのルックアップ・テーブルが再び作成された後、ブ

ログラムは第7図のユーザー・パケット切換え手順の工程190に戻る。

第10図は第9図に概略的に示した手順に従って再び作成されたルックアップ・テーブルを概念的に示す。この場合、節D及びEの間の線e1(第2図参照)が故障しており、節Dは、情報パケットを受取ったと云う応答を節Eから得ることが出来ない。この為、節A乃至Iのテーブルが払拭され、第10図に示す様にテーブルが再び作成される。丸で囲んだ項目は変更された項目であり、線の故障の結果として出来る値を示す。

第11図は、節全体、今の場合は第2図の節Eが故障した場合に起こることを概念的に示している。この場合、テーブルを再び作成する時、Eから節距離パケットが出て行くことが出来ないから、残りの節の再び作成されたテーブルは、節Eに関する項目を全く持たない。節Eは実効的に回路網から取除かれている。節Eに局部的に接続された装置を使うことを必要とする実効プログラムはつづれる。然し、節Eの局所的な装置が行なう機能

えの数)を増加する。前に述べた様に、ここで説明した経路選択手順及び節アーキテクチャは、ハイパートロイダル形アーキテクチャ以外の回路網に対して使うことが出来る。必要なことは、節が同じ数の入力及び出力を持つこと、及びそれらの接続線が全2重形式であることである。

ここで説明した基本的なアルゴリズムに種々の改良を加えることが出来る。例えば、各々の出力線に対して1つずつ、一組の伝送キューをユーザー・パケット切換え手順の一部分として設定することが出来る。この時、ユーザー・パケットは、システムの各サイクルではなく、1回だけ伝送の優先順位が割当てられる。然し、この様なキューは、他の改良と共に、アルゴリズム並びにシステムのオーバーヘッドの複雑さを強める傾向を持つ。ここで説明した経路選択手順及び節アーキテクチャは、「フェール・ソフト」形システムの回路網になる。一般的に、1つ又は更に多くの線が故障すると、計算時間がある長さだけ増加するだけであり、節全体が故障しても、回路網が動作

が回路網内のどこかで復元されていると仮定すれば、回路網のハードウェアの修理を必要とせずに、プログラムを修正して再びロードすることが出来る。第10図でも第11図でも、ある節距離値(a2B、a3D)は、それらが節Eと何等直接的に関係がなくても変化することに注意されたい。これが、全てのルックアップ・テーブルの項目を再び作成することが必要な理由である。

要約すれば、多重節相互接続回路網に対する経路選択手順及び節のアーキテクチャ、特にシステム内の節の間にユーザー・パケットを効率良く切換えるハイパートロイダル形相互接続回路網に対する手順及びアーキテクチャを説明した。経路選択手順は、静的及び動的な回路網の固定が選ばれる様に構成されている。ここで使う回路網は、ユーザー・パケット及びその宛先節に対し、重力の場と似た効果を持つ。即ち、ユーザー・パケットが宛先節に近付くにつれて、このパケットは伝送の為に更に優先的に選ばれる傾向を持ち、その全体としての節間速度(サイクル当りの切換

る為に、その後でハードウェアの修理を必要としない。

本発明の図示の実施例を詳しく説明したが、特許請求の範囲によって定められた本発明の範囲内で、種々の変更を加えることが出来ることを承知されたい。

以上の説明に関連して更に下記の項を開示する。

(1) 各々複数個の出力線及びパケット記憶装置を持つ様な複数個の節を持つ相互接続回路網の中で、夫々宛先節を持つユーザー・パケットの経路を選択する方法に於て、

第1の節で多数のパケットを受取り、前記パケットを記憶し、記憶されている少なくとも1つのパケットの宛先節を感知し、

該記憶されているパケットに対し、宛先節までの節移動距離が最短である好ましい出力線を設定し、

前記パケットをこうして決定された好ましい出力線を介して伝送する工程を含む方法。

(2) 第(1)項に記載した方法に於て、バケットに対する好ましい出力線を設定する工程が第1の節で、回路網内の全ての節までの節距離のアレーを形成し、該アレーの各メンバは節間定符号及び出力線番号によって参照され、各メンバの値は参照する出力線から参照する節までの経路の節距離であり、宛先節によって参照されるアレーの全メンバからの節移動距離の内一番短いアレーの1メンバを選び、アレーの選ばれた1メンバを参照する出力線にバケットの好ましい出力線を設定することを含む方法。

(3) 第(2)項に記載した方法に於て、各々の出力線が入力線に付設されており、節距離アレーを形成する工程が、回路網の各々の節で複数個の節距離バケットを発生し、各々の節距離バケットは節間定符号及び始めは基本値に設定される距離数を含んでおり、各々の節から接続された節に対して節距離バケットを伝送し、各々の節が、接続された節から入力線を介して節距離バケットを受取り、各々の節で、受取った各々の節距離バケッ

トの距離数を定数だけ増数し、対応する節距離アレーメンバの値を増数した各々の距離バケットからの距離数に設定し、アレーの各メンバは節間定符号並びにその距離バケットを受取った入力線に付設された出力線によって参照され、各々の節から接続された節に増数した距離バケットを再び伝送し、各々の節が増数された距離バケットを受取り、各々の節で、受取った各々の増数された距離バケットの距離数を更に増数し、各々の節で、各々の距離バケットの更に増数された距離数を対応するアレー・メンバの値と比較し、その距離数が、該距離数に対応するアレー・メンバの0以外の値に等しいか又は大きい様な、更に増数された距離バケットがあれば、それを廃棄し、各々の節で、廃棄されなかった距離バケットを全ての接続された節に再び伝送し、廃棄しなかった節距離バケットが残らなくなるまで、前記増数する工程、比較する工程、廃棄する工程、設定する工程及び再び転送する工程を繰返す工程を含む方法。

(4) 第(3)項に記載した方法に於て、更に、

動作中、接続された節がユーザー・バケットを受取らなかったことを節が感知し、この様に受取らなかったことに応答して、送信側の節の節距離アレーにある全ての現在の値を消去し、受取らなかったことに応答して、送信側の節から接続された全ての節に対して不良バケットを送信し、不良バケットを受取った各々の節の節距離アレーを消去し、受信側の各々の節が更に接続された節に対して不良バケットを再び伝送すると共に、不良バケットを受取ったことに応答して、更に接続された節の節距離アレーを消去し、回路網内の全ての節距離アレーが消去されるまで、この過程を続け、節距離アレーを発生する時と同様に、節距離バケットを発生し、受取り、増数し、比較し、廃棄し、設定し且つ再び伝送することにより、各々の節の節距離アレーを再び発生する工程を更に含む方法。

(5) 何れも複数個の出力線及びバケット記憶装置を持つ複数個の節を持っていて、各々のユーザー・バケットが宛先節を持つ様な相互接続回路網におけるユーザー・バケットの経路を選択する

方法に於て、多数のバケットを第1の節で受取り、該バケットを記憶し、記憶されている少なくともあるバケットの宛先節を感知し、記憶されているバケットの一部分の各々に対する好ましい出力線を設定し、各々の好ましい出力線は夫々の宛先節までの最短の節移動距離に関連しており、記憶されているバケットの前記部分を夫々の好ましい出力線を介して伝送する工程を含む方法。

(6) 第(5)項に記載した方法に於て、伝送される各々のバケットに対して1つの出力線だけを割当てる方法。

(7) 第(6)項に記載した方法に於て、前記バケットの各々の部分は、該部分の残りに対し、その節移動距離の昇順で好ましい出力線が割当てられる方法。

(8) 第(5)項に記載した方法に於て、各々の節が複数個のバケット記憶リンクを持ち、各々のリンクは複数個のバケット記憶バッファを持ち、各々のリンクに入力線が接続されており、更に、受取った各々のバケットをバッファに記憶し、記

値された各々のバケットに対し、好ましい出力線及びバケットの宛先節までの関連する節移動距離を決定し、空でない少なくとも1つのパッファを持つ各々のリンクに対し、該リンク内に記憶されている全てのバケットの内で最短の節移動距離を持つバケットを選択し、第1のバケットに対する好ましい出力線が別の第1のバケットに既に割当てられていなければ、節移動距離の昇順で、選択されたバケットをその好ましい出力線に割当てる工程を更に含む方法。

(9) 第(8)項に記載した方法に於て、更に、選択されたバケットを割当てた後、出力線が割当てられないで残っている場合、割当てられた選択されたバケットの伝送後、どのリンクがまだ一杯であるかを同定し、一杯と見込まれる各々のリンクに対し、出力線が割当てられないで残っていれば、割当てられていない出力線を使いきるまで、又は一杯と見込まれるリンクが残らなくなるまで、該リンクに記憶されているバケットを割当てられていない線に割当てる工程を更に含む方法。

当てられたバケットを伝送する工程が別のバケットを受取る工程と同時に行なわれる方法。

(13) 各々の節が入力線及び出力線によって複数個の他の節に接続される様な複数個の節を持ち、該節が、夫々宛先節を持つ複数個のユーザー・バケットを受取る手段と、受取ったバケットを記憶する手段と、記憶した各々のバケットの宛先節を感知する手段と、記憶されている各々のバケットに対し、該記憶されているバケットの宛先節までの節移動距離が最短である通路に接続された好ましい出力線を決定する手段と、前記バケットを前記好ましい出力線を介して伝送する手段とを持つ様なユーザー・バケットを伝送する為の相互接続回路。

(14) 第(13)項に記載した相互接続回路に於て、記憶されているバケットを夫々好ましい出力線に割当てる手段を有し、該手段はバケットの夫々の節移動距離の昇順に従って、割当ての為にバケットを選択する相互接続回路網。

(15) 第(14)項に記載した相互接続回路網に於

(10) 第(9)項に記載した方法に於て、更に、一杯と見込まれるリンクからバケットを割当てる工程の後に、出力線が割当てられないで残っていれば、割当てられていない好ましい出力線が残らなくなるまで又は各々のリンクが割当てられたバケットを持つまで、割当てられていないバケットを夫々利用し得る好ましい出力線に割当てる工程を含む方法。

(11) 第(10)項に記載した方法に於て、更に、一杯と見込まれるリンクを同定した後、一杯と見込まれるリンクに記憶されている第2のバケットに、割当てられていない出力線を割当て、割当てられた各々の第2のバケットは割当てられていない線に等しい好ましい出力線を持ち、前記選択された第2のバケットの伝送後、どのリンクが一杯であるかを同定する工程を含む方法。

(12) 第(5)項に記載した方法に於て、記憶されているバケットが出力線に割当てられ、更に、第1の節でバケットを受取る工程がそれまでに割当てられたバケットの伝送と同時に進められ、割

て、前記割当てる手段が、任意の出力線に対して1つのバケットだけを割当てる相互接続回路網。

(16) 第(14)項に記載した相互接続回路網に於て、前記感知する手段、前記決定する手段及び前記割当てる手段がプロセッサ手段で構成されており、該プロセッサ手段が一組のバケット経路選択規則を記憶するメモリ手段を含んでいる相互接続回路網。

(17) 第(10)項に記載した相互接続回路網に於て、前記節が複数個のリンクを含み、各々のリンクは複数個のバケット記憶手段を有し、節移動距離並びに好ましい出力線の利用の可否に従ってユーザー・バケットを伝送の為に選択した後、前記経路選択規則が、各々のリンクが少なくとも1つの空のバケット記憶手段を持つ見込みがあるかどうかに従って、残っている利用し得る出力線があれば、該出力線で伝送する為にバケットを選択し、前記経路選択規則は、残っていて利用し得る出力線を介して伝送する為に、空でない見込まれるリンク内に記憶された1つ又は更に多くのバケッ

トを優先的に選択する相互接続回路網。

(18) 第(17)項に記載した相互接続回路網に於て、前記メモリ手段が任意の出力線から選った、任意の節までの節移動距離を表わすルックアップ・テーブルを含んでおり、前記リンクは夫々入力線と連絡し、各々のパケット記憶手段はユーザー・パケットを記憶し得るバッファで構成され、各々のバッファが出力線に選択的に接続可能であり、前記経路選択規則は第1の工程及び第2の工程を含む複数回の工程で、前記バッファに記憶されているユーザー・パケットを割当て、前記経路選択規則は、前記第1の工程で、1つ又は更に多くのユーザー・パケットを記憶する各々のリンクに対し、最短の節移動距離を持つ1つのユーザー・パケットを選択して第1組のパケットを構成し、前記経路選択規則は更に前記第1の工程で、前記第1のリンクにある各々のパケットに対し、該パケットの節移動距離の昇順で、その出力線が既に前記第1組内のパケットに割当てられていなければ、好ましい出力線を割当て、前記プロセッサ手段が

の節が各々の出力線に1つの距離パケットを接続された節に伝送し、各々の節がその入力線から距離パケットを受取り、各々の入力線は出力線と対になっており、該対が同じ節を接続しており、受取った各々の距離パケットの距離数を受取った側の節が1だけ増数し、次に受取った側の節は増数した距離数を前記ルックアップ・テーブルの数値項目と比較し、該項目は、前記距離パケットが送ってきた入力線と対をなす出力及び該パケットの節同定符号によって参照され、前記項目が0以外である場合、前記パケットの増数された距離数が前記項目より小さくなければ、受取った側の節は前記距離パケットを廃棄し、前記増数された距離数が前記項目より小さいか或いは前記項目が0であれば、前記受取った側の節は前記項目を前記距離数に設定し、その後、前記受取った側の節は前記距離パケットを複製して該距離パケットを全ての出力線を介して隣接する節に再び伝送する相互接続回路網。

(21) 第(20)項に記載した相互接続回路網に於

て、前記第2の工程で各々のリンクを検査して、何れかのリンクにある全てのバッファが一杯であるかどうかを決定し、前記経路選択規則は、一杯のリンクがなくなるまで又は全ての出力線がパケットに割当てられるまで、一杯のリンクの内の1つパケットを利用し得る任意の出力線に割当てる相互接続回路網。

(19) 第(18)項に記載した相互接続回路網に於て、前記節のルックアップ・テーブルが動作の始めに初期設定され、各々の節は、回路網内の他の各々の節から発信されて受取った距離パケットに基づいてそのルックアップ・テーブルを作成し、各々の距離パケットは発信側の節を同定する節同定符号と発信側の節及び受取る側の節の間でその距離パケットが移動する節距離数を持っている相互接続回路網。

(20) 第(19)項に記載した相互接続回路網に於て、前記ルックアップ・テーブルの初期設定の間、各々の節は複数個の距離パケットを発信し、各々の距離パケットは始めは0の節距離数を持ち、各々

て、各々の節は該節から接続された節に伝送されたパケットを接続された節が受取らなかったことを通知することが出来、通知した節は受取らなかったことを通知したことに応答してそのルックアップ・テーブルを消去し、前記通知した節は前記受取らなかったことに応答して、接続された節に対する各々の出力線を介して不良パケットを伝送し、前記接続された節はそのルックアップ・テーブルを消去すると共に前記不良パケットを再び伝送して、回路網内の全ての節がそのルックアップ・テーブルを消去する様にし、各々の節は前記ルックアップ・テーブルを始めに作成するのと同様に、消去したルックアップ・テーブルを再生する相互接続回路網。

(22) 第(18)項に記載した相互接続回路網に於て、各々のリンクが内部出力線に接続され、該内部出力線は何れもクロスバーに接続され、該クロスバーが複数個の外部出力線と終端し、前記プロセッサ手段がリンク内に記憶されている1つまでのパケットを選択して該リンクの内部出力線と前



配クロスバーと連絡し、前記プロセッサ手段が前記選択されたバケットを1つの外部出力線に連絡する様に前記クロスバー手段に指示する相互接続回路網。

(23) 第(22)項に記載した相互接続回路網に於て、前記外部出力線が他の部に接続された複数個の出力線及び局所的な装置のインターフェースに接続された局所出力線を含んでおり、前記プロセッサ手段はリンクに記憶されている1つまでのユーザー・バケットを局所出力線に連絡する様に前記クロスバー手段に指示し、該バケットは該バケットの現在位置に等しい宛先節を持ち、前記リンクが前記装置のインターフェースから局所バケット入力経路を受取る様に選択的に接続されており、前記プロセッサ手段が局所バケットを受取って記憶する為に1つまでのバッファを選択する相互接続回路網。

(24) 第(22)項に記載した相互接続回路網に於て、各々のリンクが前記プロセッサ手段からの指令にตอบสนองして、バケットを受取る為に1つのバッ

ファを選択する為のデマルチプレクサを含み、該デマルチプレクサの入力には前記入力線の内の1つが接続されており、更に各々のリンクが、前記プロセッサ手段からの指令にตอบสนองしてバッファに記憶されているバケットを前記クロスバーに伝送する為に前記バッファの内の1つを選択するマルチプレクサを含み、該マルチプレクサは内部入力線に接続された出力端子を持っており、前記プロセッサ手段が各々のバッファに接続されて、該バッファからデータを読み取並びに該バッファにデータを蓄込むことが出来る様になっている相互接続回路網。

(25) 第(13)項に記載した相互接続回路網に於て、該回路網がハイパートロイダル形である相互接続回路網。

(26) 第(25)項に記載した相互接続回路網に於て、前記ハイパートロイダル形回路網が2リング次元に接続されている相互接続回路網。

(27) 第(25)項に記載した相互接続回路網に於て、前記ハイパートロイダル形回路網が3リング

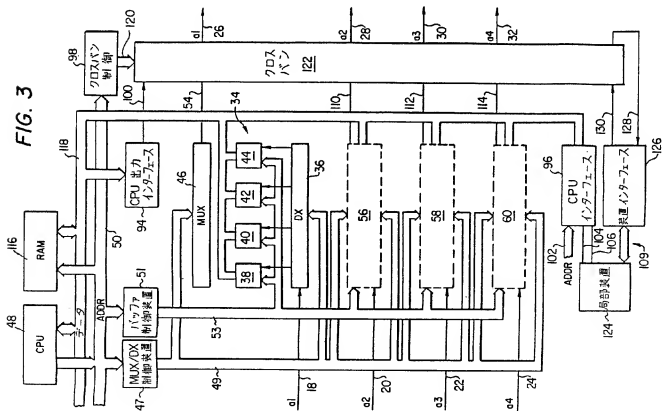
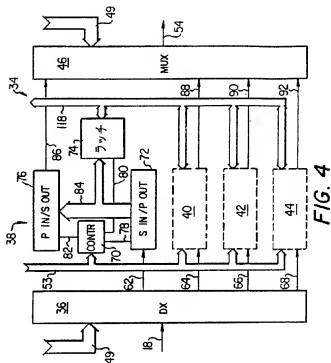
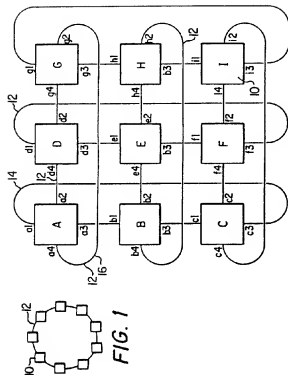
次元に接続されている相互接続回路網。

#### 4. 図面の簡単な説明

第1図は1次元のハイパートロイダル形多重節システム相互接続を示す回路図、第2図は2次元ハイパートロイダル形多重節システムの相互接続を示す回路図、第3図は第2図の1つの節の内部構造を示す回路図、第4図は第3図の横部を示す回路図で、リンクの内部構造を示している。第5図は本発明の1実施例のルックアップ・テーブル作成手順を示すフローチャート、第6図は第2図の節Aに対するルックアップ・テーブルを概念的に示す表、第7図は本発明の1実施例のユーザー・バケット切換え手順を示すフローチャート、第8図は第2図の節Aを概念的に示す表で、全てのバッファが一杯になっていることを示す。第9図は本発明の故障検出及びテーブル再作成手順のフローチャート、第10図は第2図の節Aに対するルックアップ・テーブルを概念的に示す表で、D-Eリンクの故障による項目の変化を示す。第11図は第2図の節Aに対する経路選択表を概念的

的に示す表であって、節Eの故障によるテーブルの変化を示す。

代理人 浅 村 皓



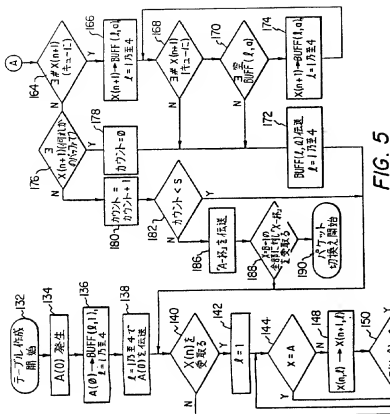
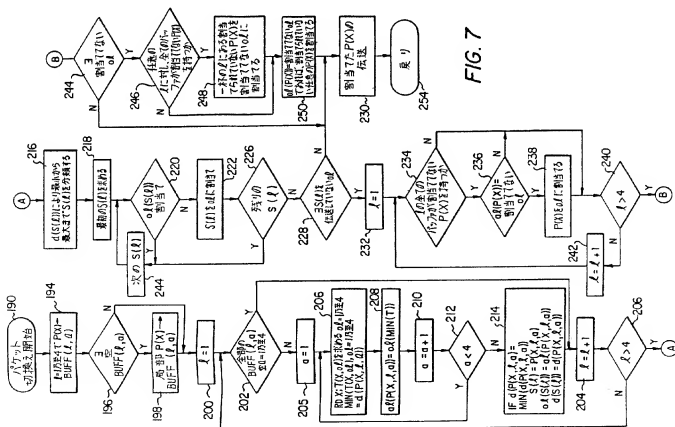
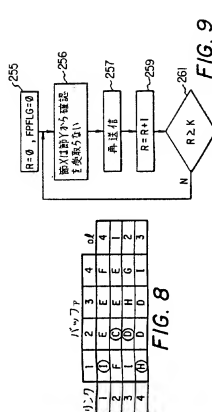


FIG. 6 -192

	B	C	D	E	F	G	H	I
a1	2	1	3	3	2	3	3	2
a2	3	3	1	2	2	2	3	3
a3	1	2	3	2	3	3	2	3
a4	3	3	2	3	3	1	2	2





**FIG. 10**

	B	C	D	E	F	G	H	I
a1	2	1	3	2	3	3	2	3
a2	0	3	1	3	2	3	3	3
a3	1	2	0	2	3	3	2	3
a4	3	3	2	3	3	1	2	2

**FIG. 11**

	B	C	D	E	F	G	H	I
a1	2	1	3	2	3	3	2	3
a2	0	3	1	3	2	3	3	3
a3	1	2	0	2	3	3	2	3
a4	3	3	2	3	3	1	2	2